Utrecht, 26 oktober 2011

Fast iterative solvers

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

with $\mathbf{A}$ $n \times n$ non-sigular.

Today's topic. Iterative methods for general systems using short recurrences

Program

- **CG**
- **Bi-CG**
- Bi-Lanczos
- Hybrid **Bi-CG**
- **Bi-CGSTAB**, **BiCGstab**(ℓ)
- **IDR**

[Hestenes Stiefel '52]

$\mathbf{A}^* = \mathbf{A} > 0$, Conjugate Gradient

$\mathbf{x} = \mathbf{0}$, $\mathbf{r} = \mathbf{b}$, $\mathbf{u} = \mathbf{0}$, $\rho = 1$

While $\|\mathbf{r}\| > tol$ do

$\sigma = -\rho$, $\rho = \mathbf{r}^*\mathbf{r}$, $\beta = \rho/\sigma$

$\mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$, $\mathbf{c} = \mathbf{A}\mathbf{u}$

$\sigma = \mathbf{u}^*\mathbf{c}$, $\alpha = \rho/\sigma$

$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$

end while

Construction CG.

There are three alternative derivations of **CG**.

- **GCR** $\rightsquigarrow$ **CR**,
  use $\mathbf{A}^{-1}$ inner product $+$ efficient implementation.
- Lanczos $+$ $T = LU$ $+$ efficient implementation.
- Orthogonalize residuals. [Ex.8

Conjugate Gradients, $\mathbf{A}^* = \mathbf{A}$, $\mathbf{K}^* = \mathbf{K}$

$$\boxed{\begin{array}{l} \mathbf{u}_k = \mathbf{K}^{-1}\mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k \end{array}}$$

Theorem. • $\mathbf{r}_k$, $\mathbf{K}\mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$

- $\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}$ is a **Krylov basis** of $\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$
- If $\mathbf{r}_k$, $\mathbf{A}\mathbf{u}_k \perp \mathbf{K}^{-1}\mathbf{r}_{k-1}$, then $\mathbf{r}_k$, $\mathbf{A}\mathbf{u}_k \perp \mathbf{K}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$

Conjugate Gradients, $\mathbf{A}^* = \mathbf{A}$, $\mathbf{K}^* = \mathbf{K}$

$$\boxed{\begin{array}{l} \mathbf{u}_k = \mathbf{K}^{-1}\mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1} \\ \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k \end{array}}$$

Theorem. • $\mathbf{r}_k$, $\mathbf{K}\mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$

- $\mathbf{r}_0, \ldots, \mathbf{r}_{k-1}$ is a **Krylov basis** of $\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$
- If $\mathbf{r}_k$, $\mathbf{A}\mathbf{u}_k \perp \mathbf{K}^{-1}\mathbf{r}_{k-1}$, then $\mathbf{r}_k$, $\mathbf{A}\mathbf{u}_k \perp \mathbf{K}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$

Proof.

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\mathbf{K}^{-1}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathbf{K}^{-1}\mathbf{r}_{k-1}$ by construction β_{k-1}

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\mathbf{K}^{-1}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathbf{K}^{-1}\mathcal{K}_{k-1}(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$ by induction:

$$\mathbf{A}\mathbf{K}^{-1}\mathbf{r}_k \perp \mathbf{K}^{-1}\mathcal{K}_{k-1}(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0) \quad \Leftrightarrow \quad \mathbf{r}_k \perp \mathbf{K}^{-1}\mathbf{A}\mathbf{K}^{-1}\mathcal{K}_{k-1}(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$$
$$\Leftarrow \quad \mathbf{r}_k \perp \mathbf{K}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1}, \mathbf{r}_0)$$

$\mathbf{A}^* = \mathbf{A}$ & $\mathbf{K}^* = \mathbf{K}$: Preconditioned CG

```
x = 0, r = b, u = 0, ρ = 1
While ‖r‖ > tol do
   Solve Kc = r for c
   σ = −ρ,  ρ = c*r,  β = ρ/σ
   u ← c − β u,  c = Au
   σ ← u*c,  α = ρ/σ
   r ← r − α c
   x ← x + α u
end while
```

Properties CG

Pros

- **Low costs per step**: 1 MV, 2 DOT, 3 AXPY
  to increase dimension Krylov subspace by one.
- **Low storage**: 5 large vectors (incl. $\mathbf{b}$).
- **Minimal res.** method if $\mathbf{A}$, $\mathbf{K}$ pos. def.: $\|\mathbf{r}_k\|_{\mathbf{A}^{-1}}$ is min.
- **Orthogonal residual** method if $\mathbf{A}^* = \mathbf{A}$, $\mathbf{K}^* = \mathbf{K}$:
  $$\mathbf{r}_k \perp \mathbf{K}^{-1}\mathcal{K}_k(\mathbf{A}\mathbf{K}^{-1};\mathbf{r}_0).$$
- No additional knowledge on properties of $\mathbf{A}$ is needed.
- **Robust**: **CG** always converges if $\mathbf{A}$, $\mathbf{K}$ pos. def..

Cons

- May **break down** if $\mathbf{A}^* = \mathbf{A} \not> 0$.
- Does **not** work if $\mathbf{A} \neq \mathbf{A}^*$.
- ◦ **CG** is sensitive to evaluation errors
  (often loss of super-linear convergence).

For general square non-singular A

- Apply **CG** to normal equations ($\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$) $\rightsquigarrow$ **CGNE**
- Apply **CG** to $\mathbf{A}\mathbf{A}^*\mathbf{y} = \mathbf{b}$ (then $\mathbf{x} = \mathbf{A}^*\mathbf{y}$)
  $\rightsquigarrow$ **Graig's method**

Disadvantage. Search in $\mathcal{K}_k(\mathbf{A}^*\mathbf{A},\ldots)$.

If $\mathbf{A} = \mathbf{A}^*$ then convergence is determined by $\mathbf{A}^2$: condition number squared, ...

[Faber Manteufel

Theorem. For general square non-singular $\mathbf{A}$, there is no Krylov solver that finds the best solution in de Krylov subspace $\mathcal{K}_k(\mathbf{A},\mathbf{r}_0)$ using short recurrences.

Alternative. Construct residuals in a sequence of shrinking spaces (orthogonal to a sequence of growing spaces): adapt the construction of **CG**.

Bi-Conjugate Gradients

$$\boxed{\begin{aligned}\mathbf{u}_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}\\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k\end{aligned}}$$

Theorem. We have $\mathbf{r}_k,\ \mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A},\mathbf{r}_0)$.

Suppose $\tilde{\mathbf{r}}_0,\ldots,\tilde{\mathbf{r}}_{k-1}$ is a **Krylov basis** of $\mathcal{K}_k(\mathbf{A}^*,\tilde{\mathbf{r}}_0)$.

If $\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \tilde{\mathbf{r}}_{k-1}$, then $\mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \mathcal{K}_k(\mathbf{A}^*,\tilde{\mathbf{r}}_0)$.

Proof.

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \tilde{\mathbf{r}}_{k-1}$ by construction α_{k-1}

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_{k-1}\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A}^*,\tilde{\mathbf{r}}_0)$ by induction

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \tilde{\mathbf{r}}_{k-1}$ by construction β_{k-1}

$\mathbf{A}\,\mathbf{u}_k = \mathbf{A}\,\mathbf{r}_k - \beta_k\,\mathbf{A}\,\mathbf{u}_{k-1} \perp \mathcal{K}_{k-1}(\mathbf{A}^*,\tilde{\mathbf{r}}_0)$ by induction:

$\mathbf{A}\mathbf{r}_k \perp \mathcal{K}_{k-1}(\mathbf{A}^*,\tilde{\mathbf{r}}_0) \quad \Leftarrow \quad \mathbf{r}_k \perp \mathcal{K}_k(\mathbf{A}^*,\tilde{\mathbf{r}}_0) \supset \mathbf{A}^*\mathcal{K}_{k-1}(\mathbf{A}^*,\tilde{\mathbf{r}}_0)$

Bi-Conjugate Gradients

$$\boxed{\begin{aligned}\mathbf{u}_k &= \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}\\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k\end{aligned}}$$

$\mathbf{r}_k,\ \mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A},\mathbf{r}_0), \qquad \mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \tilde{\mathbf{r}}_{k-1}$

With $\quad \rho_k \equiv (\mathbf{r}_k,\tilde{\mathbf{r}}_k) \quad \& \quad \sigma_k \equiv (\mathbf{A}\mathbf{u}_k,\tilde{\mathbf{r}}_k)$

and, since $\quad \tilde{\mathbf{r}}_k + \bar{\theta}_k\,\mathbf{A}^*\,\tilde{\mathbf{r}}_{k-1} \in \mathcal{K}_k(\mathbf{A}^*,\tilde{\mathbf{r}}_0) \quad$ for some θ_k,

$\bar{\cdot}$ is the complex conjugate

we have that $$\alpha_k = \frac{(\mathbf{r}_k,\tilde{\mathbf{r}}_k)}{(\mathbf{A}\mathbf{u}_k,\tilde{\mathbf{r}}_k)} = \frac{\rho_k}{\sigma_k}$$

and $$\beta_k = \frac{(\mathbf{A}\mathbf{r}_k,\tilde{\mathbf{r}}_{k-1})}{(\mathbf{A}\mathbf{u}_{k-1},\tilde{\mathbf{r}}_{k-1})} = \frac{(\mathbf{r}_k,\mathbf{A}^*\tilde{\mathbf{r}}_{k-1})}{\sigma_{k-1}} = \frac{-\rho_k}{\theta_k\,\sigma_{k-1}}$$

Bi-Conjugate Gradients

$$\boxed{\begin{array}{l}\mathbf{u}_k = \mathbf{r}_k - \beta_k\,\mathbf{u}_{k-1}\\ \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\,\mathbf{A}\,\mathbf{u}_k\end{array}}$$

$\mathbf{r}_k,\ \mathbf{u}_k \in \mathcal{K}_{k+1}(\mathbf{A},\mathbf{r}_0)$, $\quad \mathbf{r}_k,\ \mathbf{A}\mathbf{u}_k \perp \tilde{\mathbf{r}}_{k-1}$

With $\quad \rho_k \equiv (\mathbf{r}_k, \bar{q}_k(\mathbf{A}^*)\tilde{\mathbf{r}}_0) \quad \& \quad \sigma_k \equiv (\mathbf{A}\mathbf{u}_k, \bar{q}_k(\mathbf{A}^*)\tilde{\mathbf{r}}_0)$

and, since $\quad q_k(\zeta) + \theta_k\,\zeta\,q_{k-1}(\zeta) \in \mathcal{P}_{k-1} \quad$ for some θ_k,

we have that $\quad \alpha_k = \dfrac{\rho_k}{\sigma_k} \quad \& \quad \beta_k = \dfrac{-\rho_k}{\theta_k\,\sigma_{k-1}}$

Selecting the initial shadow residual $\tilde{\mathbf{r}}_0$.

- Often recommended: $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$.
- Practical experience: select $\tilde{\mathbf{r}}_0$ randomly.

Exercise. **Bi-CG** and **CG** coincide

if $\mathbf{A}$ is Hermitian and $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$.

Exercise. Derive a version of **Bi-CG** that includes a preconditioner $\mathbf{K}$.

Show that **Bi-CG** and **CG** coincide

if $\mathbf{A}$ and $\mathbf{K}$ are Hermitian and $\tilde{\mathbf{r}}_0 = \mathbf{K}^{-1}\mathbf{r}_0$.

Exercise 8.2 gives an alternative derivation of **Bi-CG**.

[Fletcher 76]

Bi-CG

$\mathbf{x} = \mathbf{0}$, $\quad \mathbf{r} = \mathbf{b}$.	Choose $\tilde{\mathbf{r}}$
$\mathbf{u} = \mathbf{0}$, $\quad \rho = 1$	$\tilde{\mathbf{c}} = \mathbf{0}$
While $\|\mathbf{r}\| > tol$ do	
$\quad \sigma = -\rho$, $\quad \rho = (\mathbf{r}, \tilde{\mathbf{r}})$, $\quad \beta = \rho/\sigma$	
$\quad \mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$, $\quad \mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$,	$\tilde{\mathbf{c}} \leftarrow \mathbf{A}^*\tilde{\mathbf{r}} - \bar{\beta}\,\tilde{\mathbf{c}}$
$\quad \sigma = (\mathbf{c}, \tilde{\mathbf{r}})$, $\quad \alpha = \rho/\sigma$	
$\quad \mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$,	$\tilde{\mathbf{r}} \leftarrow \tilde{\mathbf{r}} - \bar{\alpha}\,\tilde{\mathbf{c}}$
$\quad \mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$	
end while	

Properties Bi-CG

Pros

- Usually selects good approximations from the search subspaces (Krylov subspaces).
- **Low costs per step**: 2 DOT, 5 AXPY.
- **Low storage**: 7 large vectors.
- No knowledge on properties of $\mathbf{A}$ is needed.

Cons

- ∘ Non-optimal Krylov subspace method.
- Not robust: **Bi-CG** may break down.
- **Bi-CG** is sensitive to evaluation errors (often loss of super-linear convergence).
- ∘ Convergence depends on **shadow** residual $\tilde{\mathbf{r}}_0$.
- **2 MV needed to expand search subspace by 1 vector.**
- **1 MV is by $\mathbf{A}^*$.**

Bi-Lanczos

Find coefficients α_k, β_k, $\widetilde{\alpha}_k$ and $\widetilde{\beta}_k$ such that (**bi-orthogonalize**)

$$\gamma_k \mathbf{v}_{k+1} = \mathbf{A}\mathbf{v}_k - \alpha_k \mathbf{v}_k - \beta_k \mathbf{v}_{k-1} - \ldots \perp \mathbf{w}_k, \mathbf{w}_{k-1}, \ldots$$
$$\widetilde{\gamma}_k \mathbf{w}_{k+1} = \mathbf{A}^* \mathbf{w}_k - \widetilde{\alpha}_k \mathbf{w}_k - \widetilde{\beta}_k \mathbf{w}_{k-1} - \ldots \perp \mathbf{v}_k, \mathbf{v}_{k-1}, \ldots$$

Select appropriate scaling coefficients γ_k and $\widetilde{\gamma}_k$.

Then

$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1} \underline{H}_k$ with $\underline{H}_k$ Hessenberg

$\mathbf{A}^*\mathbf{W}_k = \mathbf{W}_{k+1} \widetilde{\underline{H}_k}$ with $\widetilde{\underline{H}_k}$ Hessenberg

and $\mathbf{W}_{k+1}^* \mathbf{V}_{k+1} = D_{k+1}$ diagonal

Exercise. $T_k \equiv D_k H_k = \widetilde{H_k}^* D_k$ is tridiagonal.

Exploit $\widetilde{H_k} = D_k H_k^* D_k^*$ and tridiagonal structure: **Bi-Lanczos**

Arnoldi: $\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1} \underline{H}_k$.

If $\mathbf{A}^* = \mathbf{A}$, then $\underline{T}_k \equiv \underline{H}_k$ tridiagonal $\rightsquigarrow$ Lanczos

---

Lanczos $\quad + \quad T = LU \quad + \quad$ efficient implementation

$\rightsquigarrow$ CG

Bi-Lanczos $\quad + \quad T = LU \quad + \quad$ efficient implementation

$\rightsquigarrow$ **Bi-CG**

[Lanczos '50]

Bi-Lanczos

Select a $\mathbf{r}_0$,	and a $\widetilde{\mathbf{r}}_0$
$\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$, $\mathbf{v}_0 = \mathbf{0}$,	$\mathbf{w}_1 = \widetilde{\mathbf{r}}_0/\|\widetilde{\mathbf{r}}_0\|$, $\mathbf{w}_0 = \mathbf{0}$
$\gamma_0 = 0$, $\delta_0 = 1$,	$\widetilde{\gamma}_0 = 0$, $\widetilde{\delta}_0 = 1$
For $k = 1, 2, \ldots$ do	
$\quad \delta_k = \mathbf{w}_k^* \mathbf{v}_k$,	
$\quad \widetilde{\mathbf{v}} = \mathbf{A}\,\mathbf{v}_k$,	$\widetilde{\mathbf{w}} = \mathbf{A}^*\mathbf{w}_k$
$\quad \beta_k = \overline{\widetilde{\gamma}}_{k-1} \delta_k / \delta_{k-1}$,	$\widetilde{\beta}_k = \overline{\gamma}_{k-1} \overline{\delta}_k / \overline{\delta}_{k-1}$
$\quad \widetilde{\mathbf{v}} \leftarrow \widetilde{\mathbf{v}} - \beta_k \mathbf{v}_{k-1}$,	$\widetilde{\mathbf{w}} \leftarrow \widetilde{\mathbf{w}} - \widetilde{\beta}_k \mathbf{w}_{k-1}$
$\quad \alpha_k = \mathbf{w}_k^* \widetilde{\mathbf{v}} / \delta_k$,	$\widetilde{\alpha}_k = \overline{\alpha}_k$
$\quad \widetilde{\mathbf{v}} \leftarrow \widetilde{\mathbf{v}} - \alpha_k \mathbf{v}_k$,	$\widetilde{\mathbf{w}} \leftarrow \widetilde{\mathbf{w}} - \widetilde{\alpha}_k \mathbf{w}_k$
$\quad$ Select a $\gamma_k \neq 0$	and a $\widetilde{\gamma}_k \neq 0$
$\quad \mathbf{v}_{k+1} = \widetilde{\mathbf{v}}/\gamma_k$,	$\mathbf{w}_{k+1} = \widetilde{\mathbf{w}}/\widetilde{\gamma}_k$,
$\quad \mathbf{V}_k = [\mathbf{V}_{k-1}, \mathbf{v}_k]$,	$\mathbf{W}_k = [\mathbf{W}_{k-1}, \mathbf{w}_k]$
end while	

Bi-CG may break down

1) **Bi-Lanczos** may **break down**, i.e., a diagonal element of D_k may be zero. **Lucky** breakdown if $\mathbf{r}_k = \mathbf{0}$.
Corresponds to $\rho = 0$ in **Bi-CG**

Remedie. Look ahead.

2) **Pivot breakdown** or **LU-breakdown**: LU-decomposition may not exist. Corresponds to $\sigma = 0$ in **Bi-CG**

Remedie.
- ∘ Composite step **Bi-CG** (pivot when needed)
- ∘ Form $T = QR$ as in **MINRES** (from the beginning): simple **Quasi Minimal Residuals**

Remedie: ∘ Restart ∘ **QMR**

Note. **CG** may suffer from pivot breakdown when applied to a Hermitian, non definite matrix ($\mathbf{A}^* = \mathbf{A}$ with positive as well as negative eigenvalues):

MINRES and **SYMMLQ** cure this breakdown.

Note. Exact breakdowns are rare.
However, near breakdowns lead to irregular convergence and instabilities. This leads to

- loss of speed of convergence
- loss of accuracy

[Sonneveld '89]

Transpose-free Bi-CG

$\rho_k = (\mathbf{r}_k, \bar{q}_k(\mathbf{A}^*)\,\tilde{\mathbf{r}}_0) = (q_k(\mathbf{A})\mathbf{r}_k, \tilde{\mathbf{r}}_0)$,
$\sigma_k = (\mathbf{A}\mathbf{u}_k, \bar{q}_k(\mathbf{A}^*)\,\tilde{\mathbf{r}}_0) = (\mathbf{A}q_k(\mathbf{A})\mathbf{u}_k, \tilde{\mathbf{r}}_0)$

$\mathbf{Q}_k \equiv q_k(\mathbf{A})$

$$\text{(Bi-CG)} \left\{ \begin{array}{ll} \rho_k, & \mathbf{Q}_k\mathbf{u}_k = \mathbf{Q}_k\mathbf{r}_k - \beta_k\,\mathbf{Q}_k\mathbf{u}_{k-1}, \\ \sigma_k, & \mathbf{Q}_k\mathbf{r}_{k+1} = \mathbf{Q}_k\mathbf{r}_k - \alpha_k\,\mathbf{A}\mathbf{Q}_k\mathbf{u}_k, \end{array} \right.$$

(Pol) Compute q_{k+1} of degree $k+1$ **s.t.** $q_{k+1}(0) = 1$.
Compute $\mathbf{Q}_{k+1}\mathbf{u}_k$, $\mathbf{Q}_{k+1}\mathbf{r}_{k+1}$

Example. $q_{k+1}(\zeta) = (1 - \omega_k\zeta)q_k(\zeta) \quad (\zeta \in \mathbb{C})$.

$$\left\{ \begin{array}{ll} \omega_k, & \mathbf{Q}_{k+1}\mathbf{u}_k = \mathbf{Q}_k\mathbf{u}_k - \omega_k\,\mathbf{A}\mathbf{Q}_k\mathbf{u}_k, \\ & \mathbf{Q}_{k+1}\mathbf{r}_{k+1} = \mathbf{Q}_k\mathbf{r}_{k+1} - \omega_k\,\mathbf{A}\mathbf{Q}_k\mathbf{r}_{k+1}, \end{array} \right.$$

Transpose-free Bi-CG; Practice

Work with $\mathbf{u}'_k \equiv \mathbf{Q}_k\mathbf{u}_k^{\rm BiCG}$ and $\mathbf{r}'_k \equiv \mathbf{Q}_k\mathbf{r}_{k+1}^{\rm BiCG}$
Write $\mathbf{u}_{k-1}$ and $\mathbf{r}_k$, instead of $\mathbf{Q}_k\mathbf{u}_{k-1}^{\rm BiCG}$ and $\mathbf{Q}_k\mathbf{r}_k^{\rm BiCG}$, resp.

$\rho_k = (\mathbf{r}_k, \tilde{\mathbf{r}}_0)$, $\quad \sigma_k = (\mathbf{A}\mathbf{u}'_k, \tilde{\mathbf{r}}_0)$

$$\text{(Bi-CG)} \left\{ \begin{array}{lll} \rho_k = (\mathbf{r}_k, \tilde{\mathbf{r}}_0), & \mathbf{u}'_k = \mathbf{r}_k - \beta_k\mathbf{u}_{k-1}, & \\ \sigma_k = (\mathbf{A}\mathbf{u}'_k, \tilde{\mathbf{r}}_0), & \mathbf{r}'_k = \mathbf{r}_k - \alpha_k\,\mathbf{A}\mathbf{u}'_k, & \mathbf{x}'_k = \mathbf{x}_k + \alpha_k\mathbf{u}'_k \end{array} \right.$$

(Pol) Compute updating coefficients for q_{k+1}.
Compute $\mathbf{u}_k$, $\mathbf{r}_{k+1}$, $\mathbf{x}_{k+1}$

Example.

$$\left\{ \begin{array}{lll} \omega_k, & \mathbf{u}_{k+1} = \mathbf{u}'_k - \omega_k\,\mathbf{A}\mathbf{u}'_k, & \\ & \mathbf{r}_{k+1} = \mathbf{r}'_k - \omega_k\,\mathbf{A}\mathbf{r}'_k, & \mathbf{x}_{k+1} = \mathbf{x}'_k + \omega_k\mathbf{r}'_k \end{array} \right.$$

Example. $q_{k+1}(\zeta) = (1 - \omega_k\zeta)q_k(\zeta) \quad (\zeta \in \mathbb{C})$.

How to choose ω_k?

Bi-CGSTABilized. With $\mathbf{s}_k \equiv \mathbf{A}\mathbf{r}'_k$,

$$\omega_k \equiv \text{argmin}_\omega \|\mathbf{r}'_k - \omega\,\mathbf{A}\mathbf{r}'_k\|_2 = \frac{\mathbf{s}_k^*\mathbf{r}'_k}{\mathbf{s}_k^*\mathbf{s}_k}.$$
.

[van der Vorst 92]

BiCGSTAB

$\mathbf{x}=\mathbf{0},\quad \mathbf{r}=\mathbf{b}.$ Choose $\tilde{\mathbf{r}}$
$\mathbf{u}=0,\quad \omega=\sigma=1.$
While $\|\mathbf{r}\|>tol$ do
- $\sigma\leftarrow-\omega\sigma,\quad \rho=(\mathbf{r},\tilde{\mathbf{r}}),\quad \beta=\rho/\sigma$
- $\mathbf{u}\leftarrow\mathbf{r}-\beta\,\mathbf{u},\quad \mathbf{c}=\mathbf{A}\mathbf{u}$
- $\sigma=(\mathbf{c},\tilde{\mathbf{r}}),\quad \alpha=\rho/\sigma$
- $\mathbf{r}\leftarrow\mathbf{r}-\alpha\,\mathbf{c},$
- $\mathbf{x}\leftarrow\mathbf{x}+\alpha\,\mathbf{u}$
- $\mathbf{s}=\mathbf{A}\mathbf{r},\quad \omega=(\mathbf{r},\mathbf{s})/(\mathbf{s},\mathbf{s})$
- $\mathbf{u}\leftarrow\mathbf{u}-\omega\,\mathbf{c}$
- $\mathbf{x}\leftarrow\mathbf{x}+\omega\,\mathbf{r}$
- $\mathbf{r}\leftarrow\mathbf{r}-\omega\,\mathbf{s}$

end while

Hybrid Bi-CG or product type Bi-CG

$$\boxed{\mathbf{r}_k\equiv q_k(\mathbf{A})\mathbf{r}_k^{\text{Bi-CG}}=q_k(\mathbf{A})\,p_k^{\text{BiCG}}(\mathbf{A})\,\mathbf{r}_0}$$

p_k^{BiCG} is the kth "**Bi-CG** residual polynomial"

How to select q_k**??**

q_k for **efficient steps** & **fast convergence**.

Fast convergence by

- reducing the residual
- stabilizing the **Bi-CG** part
- other — when used as linear solver for the Jacobian system in a Newton scheme for non-linear equations, by reducing the number of Newton steps

Hybrid Bi-CG

Examples.

CGS	**Bi-CG** $\times$ **Bi-CG**
Bi-CGSTAB	**GCR**(1) $\times$ **Bi-CG**
GPBi-CG	2-truncated **GCR** $\times$ **Bi-CG**
BiCGstab(ℓ)	**GCR**(ℓ) $\times$ **Bi-CG**

For more details on hybrid Bi-CG, see Exercise 8.12.
For a derivation of GPBi-CG, see Exercise 8.13.

Properties hybrid Bi-CG

Pros
- Converges often twice as fast as **Bi-CG** w.r.t. # MVs:
  each MV expands the search subspace
  Bi-CG: $\mathbf{x}_k-\mathbf{x}_0\in\mathcal{K}_k(\mathbf{A};\mathbf{r}_0)$ à 2k MV.
  Hybrid **Bi-CG**: $\mathbf{x}_k-\mathbf{x}_0\in\mathcal{K}_{2k}(\mathbf{A};\mathbf{r}_0)$ à 2k MV.
- Work/MV and storage similar to **Bi-CG**.
- Transpose free.
- ◦ Explicit computation of **Bi-CG** scalars.

Cons
- Non-optimal Krylov subspace method.
- Peaks in the convergence history.
- Large intermediate residuals.
- ◦ Breakdown possibilities.

[Sonneveld '89]

Conjugate Gradients Squared

$$\mathbf{r}_k = p_k^{\text{BiCG}}(\mathbf{A})\, p_k^{\text{BiCG}}(\mathbf{A})\, \mathbf{r}_0$$

CGS exploits recurrence relations for the **Bi-CG** polynomials to design a very efficient algorithm.

Properties

\+ Hybrid **Bi-CG**.

\+ A very efficient algorithm:
  1 DOT/MV, 3.25 AXPY/MV;
  storage: 7 large vectors.

– Often high peaks in its convergence history

– Often large intermediate residuals

\+ Seems to do well as linear solver in a Newton scheme

[Sonneveld 89]

Conjugate Gradients Squared

$\mathbf{x} = \mathbf{0}$, $\mathbf{r} = \mathbf{b}$. Choose $\tilde{\mathbf{r}}$.
$\mathbf{u} = \mathbf{w} = \mathbf{0}$, $\rho = 1$.
While $\|\mathbf{r}\| > tol$ do
  $\sigma = -\rho, \quad \rho = (\mathbf{r}, \tilde{\mathbf{r}}), \quad \beta = \rho/\sigma$
  $\mathbf{w} \leftarrow \mathbf{u} - \beta\, \mathbf{w}$
  $\mathbf{v} \leftarrow \mathbf{r} - \beta\, \mathbf{u}$
  $\mathbf{w} \leftarrow \mathbf{v} - \beta\, \mathbf{w}, \quad \mathbf{c} = \mathbf{A}\mathbf{w}$
  $\sigma = (\mathbf{c}, \tilde{\mathbf{r}}), \quad \alpha = \rho/\sigma$
  $\mathbf{u} = \mathbf{v} - \alpha\, \mathbf{c}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{A}(\mathbf{v} + \mathbf{u})$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\, (\mathbf{v} + \mathbf{u})$
end while

Properties Bi-CGSTAB

Pros

- Hybrid **Bi-CG**.
- Converges faster (& smoother) than **CGS**.
- More accurate than **CGS**.
- 2 DOT/MV, 3 AXPY/MV.
- Storage: 6 large vectors.

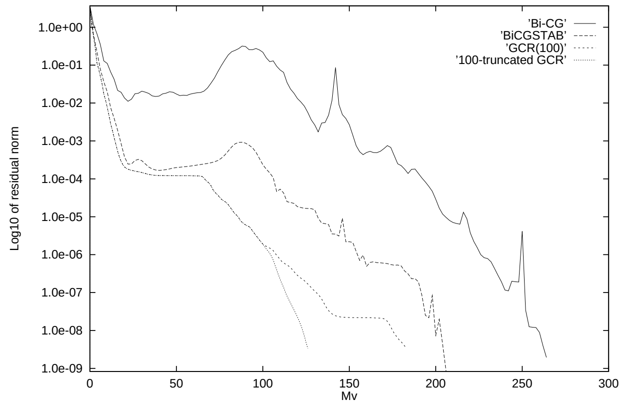
Cons

Danger of

(A) Lanczos breakdown $(\rho_k = 0)$,

(B) pivot breakdown $(\sigma_k = 0)$,

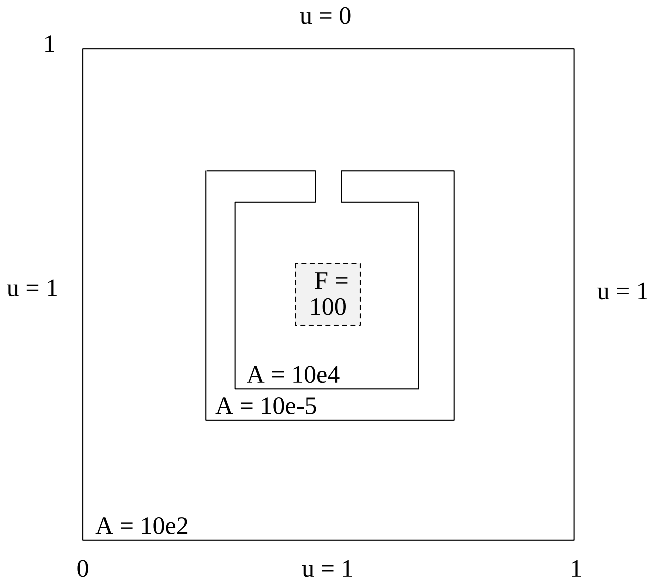
(C) breakdown minimization $(\omega_k = 0)$.



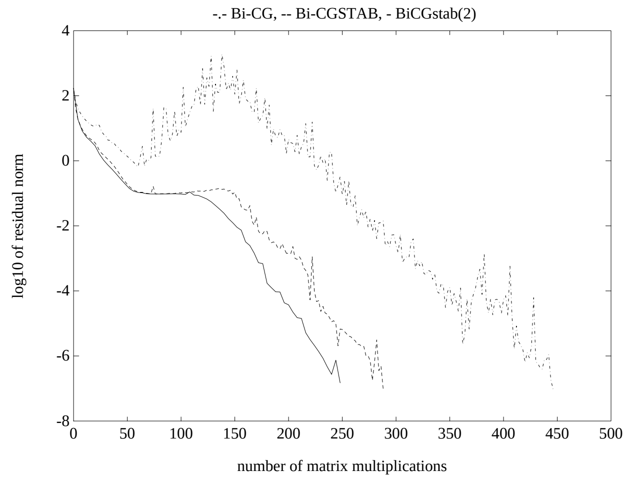
$-(a\, u_x)_x - (a\, u_y)_y = 1$ on $[0,1] \times [0,1]$.

$a = 1000$ for $0.1 \leq x, y \leq 0.9$ and $a = 1$ elsewhere.

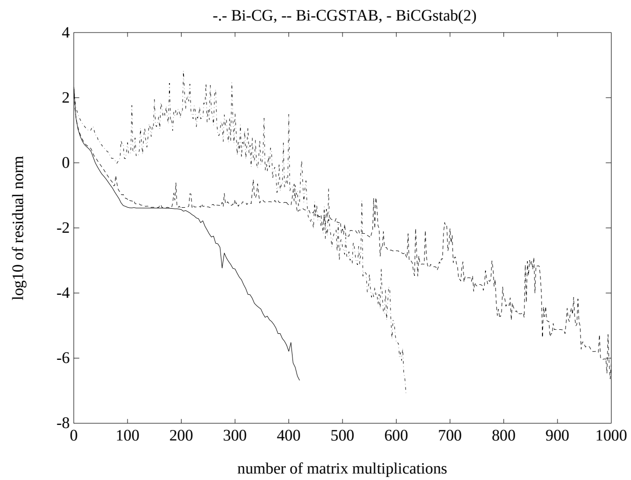
Dirichlet BC on $y = 0$, Neumann BC on other parts of Boundary.
82×82 volumes. ILU Decomp.



$-(a\,u_x)_x - (a\,u_y)_y + b\,u_x = f$ on $[0,1]\times[0,1]$.
definition of $a = A$ and of $f = F$; $F = 0$ except ...



$-(a\,u_x)_x - (a\,u_y)_y + bu_x = f$ on $[0,1]\times[0,1]$.
$b(x,y) = 2\exp(2(x^2+y^2))$, a changes strongly
Dirichlet BC. 129×129 volumes. ILU Decomp.



$-(a\,u_x)_x - (a\,u_y)_y + bu_x = f$ on $[0,1]\times[0,1]$.
$b(x,y) = 2\exp(2(x^2+y^2))$, a changes strongly
Dirichlet BC. 201×201 volumes. ILU Decomp.

Breakdown of the minimization

Exact arithmetic, $\omega_k = 0$:

— No reduction of residual by

$$\mathbf{Q}_{k+1}\, r_{k+1} = (\mathbf{I} - \omega_k \mathbf{A})\, \mathbf{Q}_k\, \mathbf{r}^{\mathrm{BiCG}}_{k+1}. \qquad (\star)$$

— q_{k+1} is of degree k: **Bi-CG** scalars can not be computed;
breakdown of incorporated **Bi-CG**.

Finite precision arithmetic, $\omega_k \approx 0$:

— Poor reduction of residual by $(\star)$

— **Bi-CG** scalars are seriously affected by evaluation errors:
drop of speed of convergence.

$\omega_k \approx 0$ to be expected if

A has eigenvalues with rel. large imaginary part: ω_k is real!

Example. $q_{k+1}(\zeta) = (1-\omega_k\zeta)q_k(\zeta) \quad (\zeta \in \mathbb{C})$.

How to choose ω_k?

Bi-CGSTABilized. With $\mathbf{s}_k \equiv \mathbf{A}\mathbf{r}'_k$,

$$\omega_k \equiv \text{argmin}_\omega \|\mathbf{r}'_k - \omega\, \mathbf{A}\mathbf{r}'_k\|_2 = \frac{\mathbf{s}_k^*\mathbf{r}'_k}{\mathbf{s}_k^*\mathbf{s}_k}.$$

BiCGstab(ℓ). Cycle ℓ times through the **Bi-CG** part to compute $\mathbf{A}^j\mathbf{u}'$, $\mathbf{A}^j\mathbf{r}'$ for $j = 0, \ldots, \ell$, where now $\mathbf{u}' \equiv \mathbf{Q}_k\mathbf{u}_{k+\ell-1}^{\text{BiCG}}$ and $\mathbf{r}' \equiv \mathbf{Q}_k\mathbf{r}_{k+\ell}^{\text{BiCG}}$ for $k = m\ell$.

$$\vec{\gamma}_m \equiv \text{argmin}_{\vec{\gamma}}\|\mathbf{r}' - [\mathbf{A}\mathbf{r}', \ldots, \mathbf{A}^\ell\mathbf{r}']\vec{\gamma}\|_2$$

$$\mathbf{r}_{k+\ell} = \mathbf{r}' - [\mathbf{A}\mathbf{r}', \ldots, \mathbf{A}^\ell\mathbf{r}']\vec{\gamma}_m$$

$$q_{k+\ell}(\zeta) = (1 - [\zeta, \ldots, \zeta^\ell]\vec{\gamma}_m)q_k(\zeta) \quad (\zeta \in \mathbb{C})$$

BiCGstab(ℓ)

$\mathbf{x} = \mathbf{0}$, $\mathbf{r} = [\mathbf{b}]$. Choose $\widetilde{\mathbf{r}}$.
$\mathbf{u} = [\mathbf{0}]$, $\gamma_\ell = \sigma = 1$.
While $\|\mathbf{r}\| > tol$ do
  $\sigma \leftarrow -\gamma_\ell\,\sigma$
  For $j = 1$ to ℓ do
    $\rho = (\mathbf{r}_j, \widetilde{\mathbf{r}})$, $\beta = \rho/\sigma$
    $\mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u}$, $\mathbf{u} \leftarrow [\mathbf{u}, \mathbf{A}\mathbf{u}_j]$
    $\sigma = (\mathbf{u}_{j+1}, \widetilde{\mathbf{r}})$, $\alpha = \rho/\sigma$
    $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{u}_{2:j+1}$, $\mathbf{r} \leftarrow [\mathbf{r}, \mathbf{A}\mathbf{r}_j]$
    $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}_1$
  end for
  $\mathbf{R} \equiv \mathbf{r}_{2:\ell+1}$. Solve $(\mathbf{R}^*\mathbf{R})\vec{\gamma} = \mathbf{R}^*\mathbf{r}_1$ for $\vec{\gamma}$
  $\mathbf{u} \leftarrow [\mathbf{u}_1 - (\gamma_1\mathbf{u}_2 + \ldots + \gamma_\ell\mathbf{u}_{\ell+1})]$
  $\mathbf{r} \leftarrow [\mathbf{r}_1 - (\gamma_1\mathbf{r}_2 + \ldots + \gamma_\ell\mathbf{r}_{\ell+1})]$
  $\mathbf{x} \leftarrow \mathbf{x} + (\gamma_1\mathbf{r}_1 + \ldots + \gamma_\ell\mathbf{r}_\ell)$
end while

BiCGstab(ℓ) for $\ell \geq 2$ [Sl Fokkema 93, Sl vdV Fokkema 94]

$$\begin{cases} q_{k+1}(\mathbf{A}) = \mathbf{A}q_k(\mathbf{A}) & k \neq m\ell \\ q_{m\ell+\ell}(\mathbf{A}) = p_m(\mathbf{A})q_{m\ell}(\mathbf{A}) & k = m\ell \end{cases}$$

where p_m of exact degree ℓ, $p_m(0) = 1$ and

$$p_m \quad \text{minimizes} \quad \|p_m(\mathbf{A})\, \underbrace{q_{m\ell}(\mathbf{A})\,\mathbf{r}_{m\ell+\ell}^{\text{BiCG}}}_{\mathbf{r}'}\|_2.$$

p_m **GCR** residual polynomial of degree ℓ

Minimization in practice: $p_m(\zeta) = 1 - \sum_{j=1}^{\ell}\gamma_j^{(m)}\,\zeta^j$

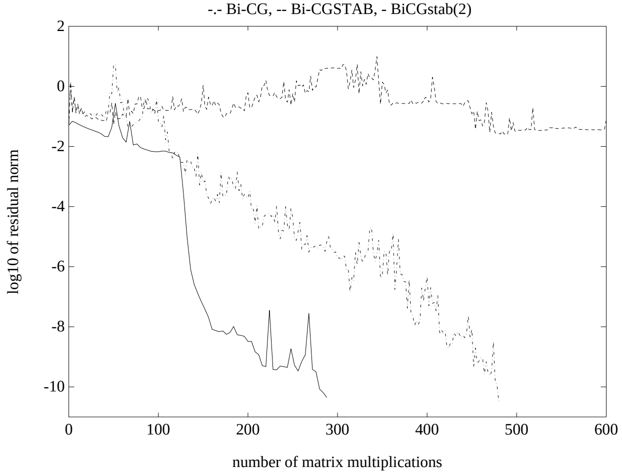
$$(\gamma_j^{(m)}) \equiv \text{argmin}_{(\gamma_j)}\|\mathbf{r}' - \sum_{j=1}^{\ell}\gamma_j\,\mathbf{A}^j\,\mathbf{r}'\|_2,$$

Compute $\mathbf{A}\mathbf{r}'$, $\mathbf{A}^2\mathbf{r}'$, $\ldots$, $\mathbf{A}^\ell\mathbf{r}'$ explicitly.

With $\mathbf{R} \equiv \left[\mathbf{A}\mathbf{r}', \ldots, \mathbf{A}^\ell\mathbf{r}'\right]$, $\vec{\gamma}_m \equiv (\gamma_1^{(m)}, \ldots, \gamma_\ell^{(m)})^{\mathsf{T}}$ we have

[Normal Equations, use Choleski] $(\mathbf{R}^*\mathbf{R})\vec{\gamma}_m = \mathbf{R}^*\mathbf{r}'$

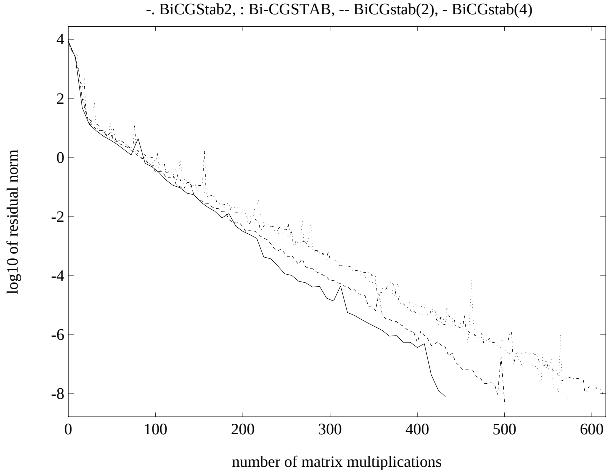
```
x = zeros(b);   rt = rand(b);
sigma = 1;   omega = 1;   u = zeros(b);
% Computation of initial residual
y = MV(x);  r = b-y;

epsilon = 10^(-16);  ell = 4;

norm = r'*r;   epsilon = norm*epsilon^2;   L = 2:ell+1;
while norm > epsilon
   sigma = -omega*sigma;  y = r;
   for j = 1:ell
        rho = rt'*y;       beta = rho/sigma;
        u = r-beta*u;
        y = MV(u(:,j));   u(:,j+1) = y;
        sigma = rt'*y;    alpha = rho/sigma;
        r = r-alpha*u(:,2:j+1);
        x = x+alpha*u(:,1);
        y = MV(r(:,j));   r(:,j+1) = y;
   end

   G = r'*r;   gamma = G(L,L)\G(L,1);   omega = gamma(ell);
   u = u*[1;-gamma];  r = r*[1;-gamma];  x = x+r*[gamma;0];
   norm = r'*r;
end
```



$u_{xx} + u_{yy} + u_{zz} + 1000 u_x = f.$

f s.t. $u(x,y,z) = \exp(xyz)\sin(\pi x)\sin(\pi y)\sin(\pi z)$.

$(52 \times 52 \times 52)$ volumes. No preconditioning.

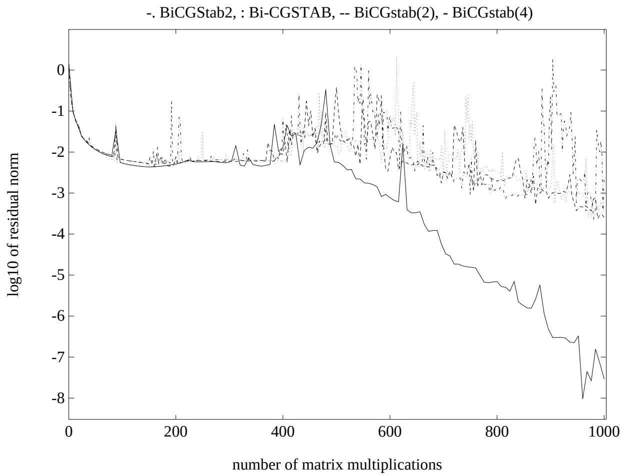


$-(a\,u_x)_x - (a\,u_y)_y = 1$ on $[0,1] \times [0,1]$.

$a = 1000$ for $0.1 \le x, y \le 0.9$ and $a = 1$ elsewhere.

Dirichlet BC on $y = 0$, Neumann BC on other parts of Boundary.

200×200 volumes. ILU Decomp.

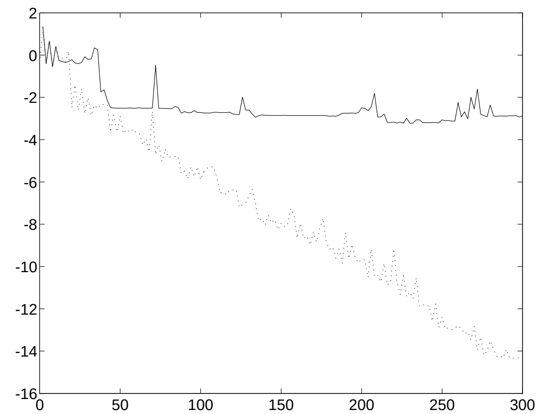


$-\epsilon(u_{xx} + u_{yy}) + a(x,y)u_x + b(x,y)u_y = 0$ on $[0,1] \times [0,1]$, Dirichlet BC

$\epsilon = 10^{-1}, \quad a(x,y) = 4x(x-1)(1-2y), \quad b(x,y) = 4y(1-y)(1-2x),$

$u(x,y) = \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y)$

(201×201) volumes, no preconditioning.



$u_{xx} + u_{yy} + u_{zz} + 1000\,u_x = f.$

f is defined by the solution

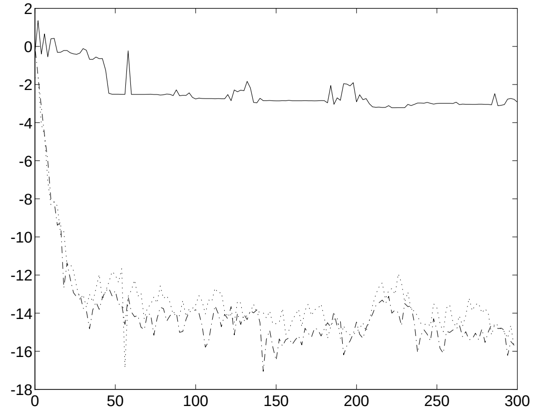
$u(x,y,z) = \exp(xyz)\sin(\pi x)\sin(\pi y)\sin(\pi z)$.

$(10 \times 10 \times 10)$ volumes. No preconditioning .

$\rho_k = (\mathbf{r}_k, \widetilde{\mathbf{r}}_0), \qquad \rho_k^\star = \rho_k(1+\epsilon)$

Accurate Bi-CG coefficients

$$|\epsilon| \leq n\,\bar{\xi}\,\frac{\|\,\mathbf{r}_k\,\|_2\,\|\,\widetilde{\mathbf{r}}_0\|_2}{|(\mathbf{r}_k,\widetilde{\mathbf{r}}_0)|} \quad = \frac{n\,\bar{\xi}}{\widehat{\rho}_k} \qquad \text{where} \qquad \widehat{\rho}_k \equiv \frac{|(\mathbf{r}_k,\widetilde{\mathbf{r}}_0)|}{\|\,\mathbf{r}_k\|_2\,\|\,\widetilde{\mathbf{r}}_0\|_2}$$



Why using pol. factors of degree ≥ 2?

Hybrid **Bi-CG**, that is faster than **Bi-CGSTAB**

1 sweep **BiCGstab**(ℓ) versus ℓ steps **Bi-CGSTAB**:

- Reduction with MR-polynomial of degree ℓ is better than $\ell\times$ MR-pol. of degr. 1.
- MR-polynomial of degree ℓ contributes only once to an increase of $\widehat{\rho}_k$

Why not?

- Efficiency:
  $1.75 + 0.25\cdot\ell$ DOT/MV, $\quad 2.5 + 0.5\cdot\ell$ AXPY/MV
  Storage: $\quad 2\ell + 5$ large vector.
- Loss of accuracy:
  $$\big|\|\mathbf{r}_k\| - \|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|\big| \leq \ldots + c\,\bar{\xi}\,\max\left(|\gamma_i|\;\big\|\,|\mathbf{A}|\,|\mathbf{A}^{i-1}\widehat{\mathbf{r}}|\,\big\|\right)$$
- break-downs are possible

Hybrid Bi-CG

Notation. If p_k is a polynomial of exact degree k, $\widetilde{\mathbf{r}}_0$ n-vector, let

$$\mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{r}}_0) \equiv \{p_k(\mathbf{A})\mathbf{v} \mid \mathbf{v} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)\}$$

Theorem. Hybrid **Bi-CG** find residuals $\mathbf{r}_k \in \mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{r}}_0)$.

Example.

Bi-CGSTAB: $\quad p_k(\lambda) = (1 - \omega_k\lambda)\,p_{k-1}(\lambda)$

where, in every step,

$\omega_k = \text{minarg}_\omega \|\mathbf{r} - \omega\mathbf{A}\mathbf{r}\|_2$, where $\mathbf{r} = p_{k-1}(\mathbf{A})\mathbf{v}$, $\mathbf{v} = \mathbf{r}_k^{\text{Bi-CG}}$

Hybrid Bi-CG

Notation. If p_k is a polynomial of exact degree k, $\widetilde{\mathbf{r}}_0$ n-vector, let

$$\mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{r}}_0) \equiv \{p_k(\mathbf{A})\mathbf{v} \mid \mathbf{v} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{r}}_0)\}$$

Theorem. Hybrid **Bi-CG** find residuals $\mathbf{r}_k \in \mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{r}}_0)$.

Example.

BiCGstab(ℓ): $\quad p_k(\lambda) = (1 - \omega_k\lambda)\,p_{k-1}(\lambda)$

where, every ℓth step

$\vec{\gamma} = \text{minarg}_{\vec{\gamma}} \|\mathbf{r} - [\mathbf{A}\mathbf{r}, \ldots, \mathbf{A}^\ell\mathbf{r}]\vec{\gamma}\|_2$, where $\mathbf{r} = p_{k-\ell}(\mathbf{A})\mathbf{r}_k^{\text{Bi-CG}}$.

$(1 - \gamma_1\lambda - \ldots - \gamma_\ell\lambda^\ell) = (1 - \omega_k\lambda)\cdot\ldots\cdot(1 - \omega_{k-\ell}\lambda)$

Induced Dimension Reduction

Definition. If p_k is a polynomial of exact degree k, $\widetilde{\mathbf{R}} \equiv \widetilde{\mathbf{R}}_0 = [\widetilde{\mathbf{r}}_1, \ldots, \widetilde{\mathbf{r}}_s]$ an $n \times s$ matrix, then

$$\mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{R}}) \equiv \left\{ p_k(\mathbf{A})\mathbf{v} \mid \mathbf{v} \perp \mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}}) \right\},$$

is the p_k-**Sonneveld** subspace. Here

$$\mathcal{K}_k(\mathbf{A}^*, \widetilde{\mathbf{R}}) \equiv \left\{ \sum_{j=0}^{k-1} (\mathbf{A}^*)^j \, \widetilde{\mathbf{R}} \, \vec{\gamma}_j \mid \vec{\gamma}_j \in \mathbb{C}^s \right\}.$$

Theorem. **IDR** find residuals $\mathbf{r}_k \in \mathcal{S}(p_k, \mathbf{A}, \widetilde{\mathbf{R}})$.

Example.

Bi-CGSTAB: $\quad p_k(\lambda) = (1 - \omega_k \lambda)\, p_{k-1}(\lambda)$

where, in every step,

$\omega_k = \text{minarg}_\omega \|\mathbf{r} - \omega \mathbf{A}\mathbf{r}\|_2$, where $\mathbf{r} = p_{k-1}(\mathbf{A})\mathbf{v}$, $\mathbf{v} = \mathbf{r}_k^{\text{Bi-CG}}$

`Select` $n \times \ell$ `matricex` $\mathbf{U}$ `and` $\widetilde{\mathbf{R}}$

Experiments suggest $\widetilde{\mathbf{R}} = \texttt{qr}(\text{rand}(n, \ell), 0)$

$\mathbf{U}$ and $\mathbf{C}$ can be constructed from ℓ steps of **GCR**.

For more details on IDR, see the Exerices 8.6, 8.14 and 8.15.

IDR

```
Select an x_0.
Select n × s matrices U and R̃.
Compute C ≡ AU.
x = x_0, r - b - Ax, j = s, i = 1
while ||r|| > tol do
    Solve R̃* C γ⃗ = R̃* r for γ⃗
    v = r - C γ⃗, s = Av
    j++, if j > s, ω = s*v/s*s, j = 0
    U e_i ← U γ⃗ + ω v, x = x + U e_i
    r_0 = r, r = v - ω s, C e_i = r_0 - r
    i++, if i > s, i = 1
end while
```